

Method of integer division or modular reduction secure
against covert channel attacks

5 The invention concerns a method of integer
division or modular reduction secure against covert
channel attacks, and in particular differential
attacks. The invention can be used for performing
division operations in a more general cryptographic
method, for example a secret or public key
10 cryptographic method. Such a cryptographic method can
for example be implemented in electronic devices such
as chip cards.

The security of cryptographic methods lies in
their ability to keep concealed the confidential data

or data derived from confidential data that they manipulate.

5 A malevolent user may possibly undertake attacks aimed as discovering in particular confidential data contained and manipulated in processing operations performed by the calculation device executing a cryptographic method.

10 Amongst the best known attacks, simple or differential covert channel attacks can be cited. Covert channel attack means an attack based on a physical quality measurable from outside the device and whose direct analysis (simple attack) or analysis according to a statistical method (differential attack) makes it possible to discover data manipulated in
15 processing operations performed in the device. These attacks have in particular been disclosed by Paul Kocher (Advances in Cryptology - CRYPTO'99, vol. 1666 of Lecture Notes in Computer Science, pp.388-397. Springer-Verlag, 1999).

20 Amongst the physical quantities that can be used for these purposes, the execution time, the current consumption, the electromagnetic field radiated by the part of the component used for executing the calculation, etc, can be cited. During the execution
25 of a method, the manipulation of a bit, that is to say its processing by a particular instruction, leaves a particular imprint on the physical quantity in question, according to the value of this bit and/or according to the instruction. In other words, the
30 trace (that is to say the change over time of the

physical quantity in question) left by the component
 executing the method is different depending on whether
 a bit equal to one or a bit equal to zero is being
 manipulated. Covert channel attacks are based on this
 5 weakness of cryptographic methods.

The cryptographic methods using as a basic
 operation a modular exponentiation operation of the
 type $Y = X^D$, X , Y and D being integer numbers, have been
 very widely studied during the past few years. By way
 10 of example, the RSA method, the key exchange according
 to Diffie-Hellman or the DSA signature method can be
 cited. Significant progress has been made in
 protecting these methods against covert channel
 attacks.

15 On the other hand, no study has been made on
 making secure cryptographic methods using as an
 elementary operation an integer division of the type q
 $= a \text{ div } b$ and $r = a \text{ mod } b$, a and b being two operands,
 q and r being respectively the quotient and the
 20 remainder of the integer division of a by b . a and/or
 b are secret data, for example elements of a key of the
 method. For example, the method of Barrett (P. Barret,
 "Implementing the RSA public key encryption algorithm
 on a standard digital signal processing", vol 263 of
 25 Lecture Notes in Computer Science, pp. 311-323,
 Springer Verlag, 1987), the method of Quisquater (US
 patent 5,166,978, November 92) or the RSA method
 implemented according to the Chinese remainder theorem
 (J J Quisquater and C Couvreur, "Fast decipherment
 30 algorithm for RSA public key cryptosystem", Electronics

Letter, vol 18, 99. 905-907, October 1982) are cryptographic methods using an integer division as an elementary operation.

5 Such cryptographic methods using an integer division method and/or a modular reduction method are sensitive to covert channel attacks, as will be seen in the following example.

10 A known method for using both an integer division and a modular reduction is the so called "paper/pencil" method. This method in practice repeats the method used when such an operation is performed by hand. This method is set out below.

15 Given two data items $a = (a_{m-1}, \dots, a_0)$ of m bits and $b = (b_{n-1}, \dots, b_0)$ of n bits, n less than or equal to m and $b_{n-1} \neq 0$, the so called "paper/pencil" division method calculates the quotient $q = a \text{ div } b$ and the remainder $r = a \text{ div } b$. For this purpose, the method successively performs several division of an integer A of $n+1$ bits by the integer b of n bits. It is
20 necessary in practice to have $0 \leq A/b < 2$, which is the case whenever $b_{n-1} \neq 0$.

The remainder r is a number of no more than n bits since $r < b$. The quotient q for its part is a number of no more than $m-n+1$ bits since $q = a \text{ div } b$ [$a \text{ div } (b_{n-1} \cdot 2^{n-1}) = a \text{ div } 2^{n-1} = (a_{m-1}, \dots, a_{n-1})$ since $b \leq b_{n-1} \cdot 2^{n-1}$ and $(a_{m-1}, \dots, a_{n-1})$ is a number of $m-n+1$ bits. At
25 the end of the division method, the quotient q is stored in the $m-n+1$ least significant bits of the register containing initially the number a . The most
30 significant bit of the remainder r is stored in 1-bit

register used as a carry during the calculation and the $n-1$ least significant bits of the remainder r are stored in the $n-1$ most significant bits of the register initially containing the number a .

5 As this work is carried out in base 2, the quotient bit of the integer division $A \text{ div } b$ has only two possible values: 0 or 1. Thus a simple way of performing the operation $A \text{ div } b$ consists of subtracting b from A and then testing the result: if
10 the result of $A - b$ is positive, then $A \text{ div } b = 1$, if the result of $A - b$ is strictly negative, then $A \text{ div } b = 0$.

The complete division method can then be written in the following manner:

15 Input: $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$
 Output: $q = a \text{ div } b$ and $r = a \bmod b$
 $A = (0, a_{m-1}, \dots, a_{m-n+1})$
 For $j = 1$ to $(m-n+1)$, do:
20 $a \leftarrow \text{SHL}_{m-1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 $A \leftarrow \text{SUB}_n(A, b)$; $\sigma \leftarrow \sigma \text{ OR } \text{carry}$
 if $(\neg \sigma = \text{TRUE})$ then $A \leftarrow \text{ADD}_n(A, b)$
 if not $\text{lsb}(a) = 1$
 End For

25 Method 1

In this method, and throughout the following, the following notations are used.

Speaking inaccurately, but in particular for reasons of clarity, and unless explicitly specified
30 otherwise, the same name will be used for speaking of a

register and its content. Thus register A will be spoken of in order to speak of the register containing the data item A.

5 The symbol " \leftarrow " and the notation $y \leftarrow x$ are used to indicate the loading of the content of the register x into a register y, the content of which is also referred to as y.

10 A is a word of n bits corresponding to the content of the n most significant bits of the register initially containing the data item a. A is of course modified at each iteration, just like the register initially containing a.

15 σ indicates whether or not the subtraction has been performed wrongly (i.e. whether the quotient bit must be equal to 0 or to 1).

$\neg\sigma$ is the complement to 1 (also referred to as negation) of the variable σ . TRUE is a constant, equal to 1 in one example.

20 $\text{lsb}(a)$ is the lowest weight bit of the number a, also referred to as the least significant bit of a.

25 $\text{SHL}_{m+1}(a, 1)$ is an operation of shifting to the left by 1 bit in the register of m+1 bits containing the data item a, the bit leaving the register being stored in the variable carry and a bit equal to 0 being entered as the least significant bit of the register initially containing the data a.

30 $\text{ADD}_n(A, b)$ is an operation of addition of the n bits of the number b to the n bits of the word A. It will be noted that the operation $\text{SHL}_n(a, 1)$ is equivalent to the operation $\text{ADD}_n(a, a)$. Naturally the

addition $ADD_n(a, b)$ is performed by adding, in an appropriate addition circuit, the content of the two registers containing respectively A and b .

5 $SUB_n(A, b)$ is an operation of subtraction of the number b from the word A . Naturally the subtraction $SUB_n(A, b)$ is performed by subtracting, in an appropriate circuit, the content of a register containing the data item b from the content of the register containing the word A .

10 In summary, the method 1 performs the following steps:

 - if $a \leftarrow SHL_{m+1}(a, 1)$ generates a carry ($\sigma = \text{carry} = 1$), this means that $a_m = 1$ (before shifting) and therefore that b must be subtracted from A .

15 - if $a_m = 0$ (before shifting) and if $A \leftarrow SUB_n(A, b)$ generates a carry ($\text{carry} = 1$), this means that $A - b \geq 0$ before subtraction and therefore b must be subtracted from A .

20 - if $a \leftarrow SHL_{m+1}(a, 1)$ does not generate a carry and if $A \leftarrow SUB_n(A, b)$ also does not generate a carry (that is to say if, after updating σ , σ is false (or $\neg\sigma$ is TRUE, then this means that $A - b < 0$ before subtraction and therefore that b would not have to be subtracted from A . In this case, the method performs
25 an addition operation $A \leftarrow ADD_n(A, b)$ in order to restore the value of A .

 The method 1 is sensitive to covert channel attacks. This is because, it is noted on method 1 that, at each iteration, according to the value of σ ,

that is to say according to the value of the quotient bit which will be obtained during the current iteration, either an addition $ADD_n(A, b)$ or a setting to 1 of the least significant bit of the register containing the data then a is performed. The use and duration of execution of these two operations are different and the trace that they leave during their implementation is also different. The overall trace left during an iteration therefore varies according to the result bit obtained during the said iteration. By measuring and studying for example the trace left by the component when the complete method is executed, it is then possible to determine bit by bit the value of the result bits.

Method 1 makes it possible to obtain both the result of the integer division ($q = a \text{ div } b$) and the remainder of the integer division ($r = a \text{ mod } b$), which is also the result of a modular reduction. Other known methods having the same drawbacks perform either a modular division alone, or a modular reduction alone. In general terms, a division method is fairly similar to a modular reduction method.

One aim of the invention is to make secure a method of implementing a division and/or a modular reduction.

For this purpose the invention proposes a cryptographic method during which there is performed an integer division of the type $q = a \text{ div } b$ and/or a modular reduction of the type $r = a \text{ mod } b$, with q a quotient, a a number of m bits, b a number of n bits, n

less than or equal to m and b_{n-1} non zero, b_{n-1} being the most significant bit of the number b .

According to the invention, the method is characterised in that the number a is masked by a random number ρ before forming the integer division and/or the modular reduction.

The number a being masked by a random number, the trace (for example the energy consumption) left during the execution of the method is different at each execution, so that it is no longer possible to implement a differential covert channel attack.

The invention can be applied for example to method 1, which performs both a division and a modular reduction. The invention can more generally be applied to any method which performs one or other of these operations.

The random number ρ can be modified at each execution of the method, or simply after a predefined number of executions of the method. Where applicable, the said predefined number is preferably chosen relatively small, for example a number from 32 to 64 bits.

According to a preferred embodiment of the invention, in order to mask the number a , there is added, to the number a , b times the random number ($a \leftarrow a + b \cdot \rho$). For this purpose, in concrete terms, the content of the register b is multiplied by the random number ρ and then added to the number a , and the result of the addition is then stored in the register initially containing the number a .

Then the integer division and/or the modular reduction required is next performed.

5 In the case where an integer division is performed, the result of the integer division performed with the number a masked in the form $a + b \cdot p$ is equal to $a \text{ div } b + p$. In this case, after the integer division, the contribution made by the random number p in order to find the expected result of the integer division on the number a , that is to say $a \text{ div } b$, is
10 taken away from the result of the integer division.

In the case where modular reduction is performed, the result of the operation $(a + b \cdot p) \bmod b$ is equal to $a \bmod b$, the expected result of the modular reduction on the number a .

15 The invention also concerns an electronic component comprising means for implementing a method according to the invention, as described above. The programmed calculation means comprise in particular several registers for storing the numbers a and b .

20 Finally, the invention concerns a chip card comprising a component having the characteristics described above.